(2)

AD-A216 642

SAM AND IVAN: AUTOMATED AGENTS FOR ANALYTIC WAR GAMING

William Schwabe

May 1988

P-7443

The RAND Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

SAM AND IVAN: AUTOMATED AGENTS FOR ANALYTIC WAR GAMING[1]

A recent Briefing article in *Jane's Defence Weekly*[2] discussed Pentagon war gaming and the RAND Strategy Assessment System (RSAS). Readers may have been left with the erroneous impression that the RSAS has been widely used in government for war gaming and contingency planning. That is not yet true. RSAS has been delivered to the Office of the Secretary of Defense, the Organization of the Joint Chiefs of Staff, the National Defense University, and elsewhere, but it is a complex tool with much refinement still needed, and those organizations are just beginning to learn to use it.

The *Jane's* article described a game focused on the Persian Gulf, which was actually a series of analytic cases studied in 1982.[3] Readers of the article have asked whether that's really how the RSAS works. I will try to answer that question here. To do that, I'll first compare RSAS capabilities in 1982 and now. I'll then describe the purposes, procedures, and outcomes of the games described in the *Jane's* article, noting how a study of similar problems using today's RSAS would be undertaken. Finally, I'll try to put the use of automated war gaming into a realistic perspective.

RSAS THEN AND NOW

The first version of the RSAS was developed in 1980 as a demonstration of concept, the concept being that, by giving computer programs a game-like structure, the contextual richness of war gaming could be combined with the explicitness and replicability of computer simulation. The "game-like" structure meant having Red Agent programs play against Blue Agent programs, with game control functions being performed by a Scenario Agent, representing

[1]This paper responds to and amplifies information about the RAND Strategy Assessment System (RSAS) given in a recent article in *Jane's Defence Weekly*. It contrasts today's RSAS with earlier versions, and it discusses purposes, procedures, and outcomes of RSAS use. The author is Associate Director of RAND's Strategy Assessment Center.

[2]Thomas B. Allen, "Sam and Ivan: Bottom Line in Wargames," *Jane's Defence Weekly*, 6 February 1988, pp. 217-218.

[3]William Schwabe, *Strategic Analysis as Though Nonsuperpowers Matter*, The RAND Corporation, Santa Monica, CA, N-1997-DNA, June 1983.

nonsuperpower governments, and a Force Agent, simulating force operations and interactions. There was also the notion that the system could be run with either human teams or the computer programs playing any of the parties.

Why was this worthwhile? Because it was felt that strategic analysis, specifically strategic nuclear analysis using computer models, had become an esoteric exercise devoid of context, seldom if ever positing why the hypothetical nuclear exchange was happening. War gaming, it was widely felt, offered contextual richness, but it often lacked rigor, especially replicability. It was generally infeasible to use war gaming to compare cases, that is, results under somewhat different conditions.

The RSAS agents were and are rule-based programs, consisting largely of statements of the form: if X, then Y, else Z. In the Mark I version only Red, Scenario, and Force Agents were executable computer programs; Blue was played by human teams. Red Agent was a pattern-matching program that let the operator choose from among four sets of actions selected from a library as most nearly appropriate given the current situation.

The Mark II RSAS was developed in 1981-82 and was the version used to generate the scenarios described in the *Jane's* article. It featured both Red and Blue Agent semiautomatic computer programs, together with a more fully automated Scenario Agent. Given a situation, Red or Blue Agent displayed appropriate if-then-else rules to the operator, who made keyboard entries to implement the rules. Force Agent was a nuclear exchange model, modified to include some aspects of ground and air warfare. These models served as prototypes for subsequently developed, more fully automated Agent programs.

The current version of RSAS is release 3.0. It features Red and Blue Agents with a hierarchy of national and military command levels. The national command level models are the Sams and Ivans. Currently there are two versions of Sam, one being a coalition leader and the other more of a unilateralist, and two of Ivan, one being a pragmatic opportunist and the other a less conditional but conservative warfighter. The military command levels beneath them represent theater commanders. The programs they execute are called analytic war plans, each representing a campaign plan with various bounds and authorizations that can be set by higher authority (human player or automated Sam or Ivan). The analytic war plans have different versions, each representing a different theater strategy. Currently, the most highly developed theater is NATO's Central Region, for which

there are four Blue and ten Red plans. At present, these are only RAND's notions of campaign plans, but they are designed to be modified by analysts in government to represent policies and strategies in which they may be interested.

Today's Force Agent is far more sophisticated than in Mark II. Air-land, naval, and strategic nuclear theaters are modeled, as is strategic mobility and strategic command-control. They receive their orders directly from Red, Blue, or Green (as Scenario is now called) Agents; alternatively, they can be operated interactively by a user-analyst.

Unlike the computer program in the movie, *War Games*, these programs are in no way connected to operational command-control systems. RSAS was, for good reasons, never intended for that purpose.

PURPOSES, PROCEDURES, AND OUTCOMES

The *Jane's* article describes a baseline case scenario, produced using RSAS Mark II, in which civil war in Iran leads to a Soviet invasion of Iran, U.S. and allied intervention in Southwest Asia, and superpower conflict which spreads to Europe and which escalates to general, strategic nuclear warfare. The article is faithful to the development of the baseline scenario, but it may mislead readers into thinking that RSAS programs considered off-baseline, less cataclysmic options, and hell-bent for Armageddon, rejected them in favor of the baseline case. That's not at all how it happened (or happens with the current RSAS).

To generate the baseline scenario, the RSAS user-analyst, then and now, enters a host of assumptions expressed as rules, data, and control parameters. The Ivan and Sam models do lookaheads to select from among choices of strategies. To do this, the Ivan and Sam models themselves must make assumptions about the nature of their opponents and other countries. In the baseline case some of those lookahead assumptions were wrong (i.e., they did not match the "real" behavioral assumptions entered by the user-analyst for that case), and the result was a chain of interactions that escalated the conflict to strategic nuclear warfare, something well beyond the original objectives of either side. That happens sometimes in human player war games. The advantage of doing this with rule-based models is that it facilitates tracing the logic, so we can better understand what happened and why.

By design, the 1982 analysis posited and compared several different off-baseline cases. These cases were intended to explore ways in which the responses of nonsuperpower countries to superpower conflict might affect the outcome. They

were cases considered by the analyst using RSAS--not cases somehow considered and rejected by the RSAS. Thus, RSAS did not simply give up and deny Israel was a player. In the lingo of the Mark II RSAS, the analyst typed in the ROSIE language statement, "Deny Israel is a player" to turn off Scenario Agent's rules for Israel, so that a human team could play Israel. In today's RSAS 3.0 an analyst can do the same thing, but now it's done with a Data Editor entry telling Green Agent in the RAND-ABEL language that Israel's Player-status is No.

Why did we want to have humans play Israel? In the Saudi-Israeli game case, we had human teams play Saudi Arabia and Israel because we thought the Scenario Agent rules might not adequately reflect difficulties if the United States sought support from both Israel and Arab countries. In the game, it turned out there were no Arab-Israeli difficulties; however, an unanticipated situation did arise. The Saudi team took offense that the other Gulf Cooperation Council states, played by Scenario Agent, paid too little attention to Saudi regional leadership--so the Saudi team refused to cooperate with the United States and sat out the war.

The results of running the various analytic cases are summarized in Fig. 1.

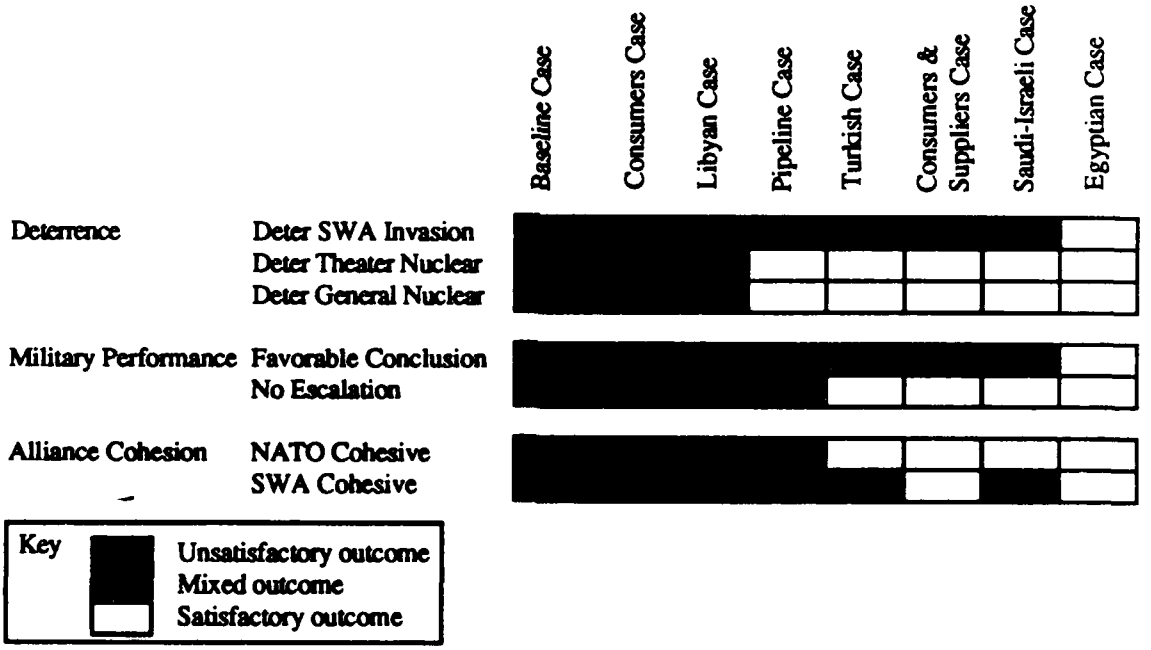

Fig. 1 -- Comparison of Case Results

Three types of measures of outcomes were considered: deterrence, military performance if deterrence failed, and alliance cohesion. The RSAS case runs suggested that outcomes could range from highly satisfactory to highly unsatisfactory, as the result solely of differences in nonsuperpower behavior. We can say "solely" here because we know these were the only factors that differed among cases; one can never be so confident of the causal relationships in human player games because the "real" reasons are inside the players' heads.

It is important to recognize that RSAS generated the case results, but RSAS did not generate the analysis nor the conclusions. With RSAS then and now, you don't just turn on the machine and let it print out an analysis. RSAS-based analysis is an iterative process with lots of off-line preliminary work, careful specification of case assumptions, refinement of data and rules, careful tracing of the Agents' logic during the runs, comparison of case results, formulation of conclusions, and further testing of the conclusions.

How are studies conducted with today's RSAS? We generally begin with group discussions to identify the scope of the problem and its issues. We then assess present RSAS capabilities to address the issues, and we may make a few preliminary system runs. This leads to a design for new theaters, functions, rules, or displays. We do the necessary research and write the computer code. During these stages of the study, the Force models are often the real workhorses, with analysts spending many hours operating them interactively with mouse and keyboard, carefully tracing cause and effect. We often conduct practice simulation runs or war games. When we are satisfied with them, we do production runs or games, compare results, formulate conclusions, and do sensitivity tests. Finally, we decide which of the new material should be added to the standard library of RSAS rules.

How does this differ from life-without-RSAS for the professional defense analyst? Because RSAS offers a library of rule sets, for example, you can have NATO simulate Simple Alert by entering "Perform Simple-alert." If you want to see how RSAS represents Simple Alert, you can look at the RAND-ABEL rules on-line and, son-of-a-gun, you can understand them even if you're not a computer programmer. And if you want to improve them, you can move a copy into a special file directory, edit it, and have the changed version interpreted and executed as soon as you restart the game run. During and after the game run you can look at on-line graphics--time plots and map displays--or logs of Agent decisions and reasons.

None of this substitutes for thinking or for integrity. RSAS is not an appliance; it's a set of power tools.

ON BALANCE

Some people fear that computers will crowd out human judgment. Are the results from RSAS runs really the "bottom line"? Of course not. In the real world the bottom line is how things turn out. In the world of war gaming, it's how things might plausibly turn out. Because RSAS permits consideration of many different cases, it is possible to consider a greater range of strategies. Because RSAS represents decision logic explicitly in its rules, it is possible for people to review the plausibility of outcomes more rigorously.

Is RSAS the "bottom line" in war games? No. It is sophisticated, cutting-edge work, but RSAS will be refined a great deal over the next few years as the result of applications. And in time something better will come along. But for now and for some time into the future, RSAS will be used by defense agencies and war colleges to gain better understanding of the pros and cons of alternative policies, strategies, and concepts of operations.